Communications in Combinatorics and Optimization
Vol. 4 No. 2, 2019 pp.201-208
DOI: 10.22049/CCO.2019.26469.1116

CCO
Commun. Comb. Optim.

On Hop Roman Domination in Trees

Nader Jafari Rad[1]*, Abolfazl Poureidi[2]

[1] Department of Mathematics, Shahed University, Tehran, Iran
n.jafarirad@gmail.com

[2] Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran
a.poureidi@gmail.com

Received: 7 April 2019; Accepted: 24 May 2019
Published Online: 26 May 2019

In honor of Lutz Volkmann on the occasion of his seventy-fifth birthday

Abstract: Let $G = (V, E)$ be a graph. A subset $S \subset V$ is a hop dominating set if every vertex outside S is at distance two from a vertex of S. A hop dominating set S which induces a connected subgraph is called a connected hop dominating set of G. The connected hop domination number of G, $\gamma_{ch}(G)$, is the minimum cardinality of a connected hop dominating set of G. A hop Roman dominating function (HRDF) of a graph G is a function $f : V(G) \longrightarrow \{0, 1, 2\}$ having the property that for every vertex $v \in V$ with $f(v) = 0$ there is a vertex u with $f(u) = 2$ and $d(u, v) = 2$. The weight of an HRDF f is the sum $f(V) = \sum_{v \in V} f(v)$. The minimum weight of an HRDF on G is called the hop Roman domination number of G and is denoted by $\gamma_{hR}(G)$. We give an algorithm that decides whether $\gamma_{hR}(T) = 2\gamma_{ch}(T)$ for a given tree T.

Keywords: hop dominating set, connected hop dominating set, hop Roman dominating function

AMS Subject classification: 05C69

1. Introduction

For notation and graph theory terminology not given here, we refer to [7]. Let $G = (V, E)$ be a graph with the vertex set $V = V(G)$ and the edge set $E = E(G)$. The *order* of G is $n(G) = |V(G)|$. The *open neighborhood* of $v \in V$ is $N_G(v) = \{u \in V(G) | uv \in E(G)\}$. The *open neighborhood* of S is $N_G(S) = \cup_{v \in S} N_G(v)$ and the *closed neighborhood* of S is $N_G[S] = N_G(S) \cup S$, where $S \subseteq V$. The *degree* of v,

* *Corresponding Author*

denoted by $\deg(v)$, is $|N_G(v)|$. The *distance* between two vertices u and v in G, denoted by $d(u,v)$, is the minimum length of a (u,v)-path in G. The diameter of G, $diam(G)$, is the maximum distance among all pairs of vertices in G. For an integer $k \geq 1$, the set of all vertices at distance k from v is denoted by $N_k(v)$. Also, we denote $N_k(v) \cup \{v\}$ by $N_k[v]$. A vertex of degree one in a tree is referred as a *leaf* and its unique neighbor as the *support vertex*. We denote the set of leaves of a tree T by $L(T)$ and the set of support vertices by $S(T)$.

Ayyaswamy and Natarajan [4] introduced the concept of hop domination in graphs. A set $S \subseteq V$ is a *hop dominating set* (HDS) if every vertex outside S is at distance two from a vertex of S. Furthermore, if S induces a subgraph of G that is connected, then S is a connected hop dominating set of G. The (*connected*) *hop domination number* of G, $(\gamma_{ch}(G))$ $\gamma_h(G)$, is the minimum cardinality of a (connected) hop dominating set of G. An HDS of G of minimum cardinality is referred as a $\gamma_h(G)$-set. The concept of hop domination was further studied in [3, 8, 10].

A function $f : V \longrightarrow \{0,1,2\}$ having the property that for every vertex $v \in V$ with $f(v) = 0$, there exists a vertex $u \in N(v)$ with $f(u) = 2$, is called a *Roman dominating function* or just an RDF. The mathematical concept of Roman domination defined and discussed by Stewart [14] and ReVelle and Rosing [11] and subsequently developed by Cockayne et al. [5]. Several variations of Roman domination have been already studied, see for example, [1, 2, 6, 15, 16].

A *hop Roman dominating function* (HRDF) is a function $f : V \longrightarrow \{0,1,2\}$ having the property that for every vertex $v \in V$ with $f(v) = 0$ there is a vertex u with $f(u) = 2$ and $d(u,v) = 2$. The weight of an HRDF f is the sum $f(V) = \sum_{v \in V} f(v)$. The minimum weight of an HRDF on G is called the *hop Roman domination number* of G and is denoted by $\gamma_{hR}(G)$. An HRDF with minimum weight is referred as a $\gamma_{hR}(G)$-function. For an HRDF f in a graph G, we denote by V_i (or V_i^f to refer to f) the set of all vertices of G with label i under f. Thus, an HRDF f can be represented by a triple (V_0, V_1, V_2) and we can use the notation $f = (V_0, V_1, V_2)$. We remark that by this time there is no polynomial algorithms for hop Roman domination number. Hop Roman domination in graphs was introduced by Shabani in [12] and further studied in [9, 13]. Assigning the value 2 to every vertex in an HDS of a graph and zero to each other vertex yeilds an HRDF, as it is observed by Shabani.

Theorem 1 (Shabani [12]). *For any graph G, $\gamma_{hR}(G) \leq 2\gamma_h(G)$.*

Since always, $\gamma_h(G) \leq \gamma_{ch}(G)$ for every graph G, we thus have $\gamma_{hR}(G) \leq 2\gamma_h(G) \leq 2\gamma_{hc}(G)$ for every graph G.

In this paper, we give an algorithm that decides whether $\gamma_{hR}(T) = 2\gamma_{ch}(T)$ for a given tree T.

Algorithm 2.1: COMPUTE-INNER-VERTICES (T)

```
Input: A tree T.
Output: The set of all inner vertices of T, i.e., I(T).
I(T) := ∅.
for each v ∈ T    Compute T_v.
if diam(T) = 4 then
| I(T) = I(T) ∪ {v};
end
return I(T) ;
```

2. Trees T with $\gamma_{hR}(T) = 2\gamma_{ch}(T)$

Let $\mathcal{T}_c$ be the set of all trees T with $\gamma_{hR}(T) = 2\gamma_{ch}(T)$. It is easy to see that the following is true.

Observation 1. *If $T \in \mathcal{T}_c$, then there is a $\gamma_{hR}(T)$-function $f = (V_0, V_1, V_2)$ with $V_1 = \emptyset$ such that V_2 induces a connected subtree of T.*

We propose an algorithm to decide whether a given tree is or not in $\mathcal{T}_c$. We first present some definitions. We say that a vertex u of a tree T is adjacent to a *hop leaf* v if v is a leaf of T with the support vertex s such that $\deg(s) = 2$ and vertices u and s are adjacent. Given a positive integer n, let T_n be a tree obtained from P_n by adding (at least) two hop leaves to any vertex of P_n, where P_n is a path graph with n vertices. It is easy to see that $T_n \in \mathcal{T}_c$. So, $\mathcal{T}_c$ is an infinite family.
Given a tree T, we say that v is an inner vertex of T if there are (at least) two distinct vertices x and y at distance 2 from v in T with $d(x, y) = 4$. Let $I(T)$ be the set of all inner vertices of T. Let $N_2'(v) = \{u \in V(T) \backslash I(T) | d(u, v) = 2\}$, and let $S_v = N_2'(v) - \cup_{u \in I(T) \backslash \{v\}} N_2'(u)$. Let T_x be the subtree of T induced by $N_2[x]$, where $x \in V(T)$. Clearly, $diam(T_x) \leq 4$. It is easy to see that the following result is true.

Observation 2. *Given a tree T, vertex v is an inner vertex of T if and only if $diam(T_v) =$ 4.*

Lemma 1. *Let T be a tree. Algorithm 2.1 computes the set of all inner vertices of T, i.e., $I(T)$, in $\mathcal{O}(|V(T)|)$ time.*

Proof. Clearly, there is an algorithm to compute T_v and the diameter of T_v in $\mathcal{O}(|V(T_v)|)$ time. We have $|V(T_v)| = 1 + \sum_{u \in N(v)} \deg(u)$. To compute $I(T)$ by Observation 2 it suffices to compute T_v for any vertex v of T and check whether $diam(T_v) = 4$. Clearly, Algorithm COMPUTE-INNER-VERTICES does this. It remains to compute the time complexity of Algorithm COMPUTE-INNER-VERTICES. Clearly, the running time of Algorithm COMPUTE-INNER-VERTICES is $\mathcal{O}(\sum_{v \in V(T)} |V(T_v)|)$. Let $V(T) = \{v_1, v_2, \dots, v_n\}$, and let $S_T = \{T_{v_1}, T_{v_2}, \dots, T_{v_n}\}$. Assume that $|S_T| =$

$|V(T_{v_1})| + \ldots + |V(T_{v_n})|$, that is, $\sum_{v\in V(T)} |V(T_v)| = |S_T|$. Let $e = xy$ be an edge of T. It is easy to see that e appears in $\deg(x) + \deg(y)$ trees of S_T. So, $|S_T| = n + \sum_{e=xy\in E(T)}(\deg(x) + \deg(y)) = n + 2\sum_{v\in V(T)} \deg(v)$. Therefore, Algorithm Compute-Inner-Vertices computes $I(T)$ in $\mathcal{O}(|V(T)|)$ time. □

Lemma 2. *Let T be a tree with $diam(T) \geq 5$. Then $T \in \mathcal{T}_c$ if and only if $|S_v| \geq 2$ for any inner vertex v of T.*

Proof. (⇒) Let T be a tree of $\mathcal{T}_c$ with $diam(T) \geq 5$, and let $v_1, v_2, \ldots, v_{diam(T)+1}$ be any longest path of T.
Assume first that $diam(T) = 5$. It is easy to see that $T \in \mathcal{T}_c$ and $I(T) = \{v_3, v_4\}$. We have $\{v_1, v_5\} \subseteq N'_2(v_3)$, both v_1, v_5 are not in $N'_2(v_4)$, $\{v_2, v_6\} \subseteq N'_2(v_4)$ and both v_2, v_6 are not in $N'_2(v_3)$. Therefore, $|S_{v_3}| = |N'_2(v_3) - N'_2(v_4)| \geq 2$ and $|S_{v_4}| = |N'_2(v_4) - N'_2(v_3)| \geq 2$. It follows that the claim holds for any tree with diameter 5.
Assume that $diam(T) = 6$. By Observation 1 there is a $\gamma_{hR}(T)$-function f with $V_1 = \emptyset$ such that V_2 induces a connected subtree of T. It is easy to see that all vertices v_3, v_4, v_5 are in V_2; otherwise the subtree of T induced by V_2 is a disconnected tree. So, $|V_2| \geq 3$. Clearly, $v_4 \in I(T)$ and both $v_2, v_6 \in N'_2(v_4)$. Also, there is no vertex of $I(T) - \{v_4\}$ at distance 2 from v_2 or v_6. It means that v_4 is the only vertex of $I(T)$ for which $N'_2(v_4)$ contains v_2 (respectively, v_6). It follows that we have $|S_{v_4}| \geq 2$.
Assume that $diam(T) \geq 7$. By Observation 1 there is a $\gamma_{hR}(T)$-function f with $V_1 = \emptyset$ such that V_2 induces a connected subtree of T. It is easy to see that $|V_2| \geq 4$. Suppose for a contradiction there is a vertex v in $I(T)$ such that $|S_v| < 2$. If $f(v) = 0$, then the subtree of T induced by V_2 is a disconnected tree. So, $f(v) = 2$. There are the following cases to consider.

- $S_v = \emptyset$.

  As mentioned in Case 1, when $diam(T) = 6$, we have $v \neq v_4$. So, there is a vertex w in V_2 (in both Cases 1 and 2) such that $d(v, w) = 2$. We replace $f(v)$ by 0 to obtain an HRDF on T with weight less than $w(f)$, a contradiction.

- $S_v = \{x\}$.

  We replace $f(v)$ by 0 and $f(x)$ by 1 to obtain an HRDF on T with weight less than $w(f)$, a contradiction.

So, if $T \in \mathcal{T}_c$ with $diam(T) \geq 5$, then $|S_v| \geq 2$ for any inner vertex v of T.
(⇐) Assume that for any inner vertex v of tree T with $diam(T) \geq 5$ we have $|S_v| \geq 2$. Since $diam(T) \geq 5$, we have $\gamma_h(T) \geq 2$. We prove that $T \in \mathcal{T}_c$ by induction on the hop domination number of T.
Base case: It is easy to see that $\gamma_h(T) = 2$ holds only for trees with the diameter 5. (Note that by the hypothesis of the claim $diam(T) \geq 5$.) We know that the claim holds for any tree with diameter 5. This proves the base case of the induction.
Induction hypothesis: Assume that all trees with $\gamma_h(T) = m \geq 2$ and with $|S_v| \geq 2$ for any inner vertex v of T are in $\mathcal{T}_c$.

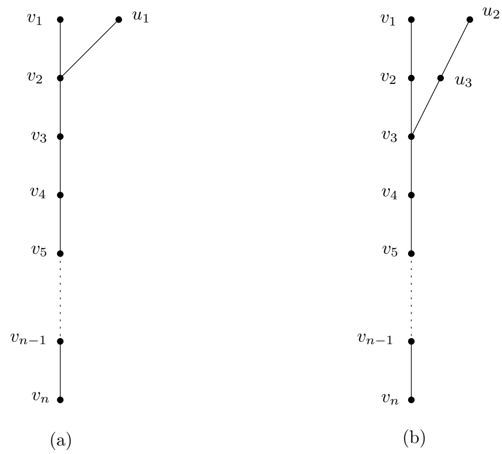

Figure 1. **For the proof of Lemma 2.**

The inductive step: Let T be a tree with $\gamma_h(T) = m+1$ such that $|S_v| \geq 2$ for any inner vertex v of T. Let $v_1, v_2, \dots, v_t$ be any longest path of T, where $t \geq 7$. Both $v_1, v_2 \notin I(T)$, but all vertices $v_3, \dots, v_{t-2} \in I(T)$, where $t-2 \geq 5$. We have $v_1 \in S_{v_3}$. Let v be any node adjacent to v_4. Clearly, if $v \in I(T)$, then $v \notin S_{v_3}$ and if $v \notin I(T)$, then $v \in N_2'(v_5)$. Therefore, $v \notin S_{v_3}$. See Figure 1. Since $|S_{v_3}| \geq 2$, either there is at least one leaf u_1 adjacent to v_2 with $u_1 \neq v_1$; see Figure 1(a) or there is one leaf u_2 with $d(v_3, u_2) = 2$ such that u_2 is not adjacent to v_2 and v_4; see Figure 1(b).
So, S_{v_3} contains only leaves of T. Let $T' = T - S_{v_3}$. Note that $diam(T') \geq 5$. It is easy to see that v_3 is not an inner vertex of T'. So, T' is a tree such that $|S_v| \geq 2$ for any inner vertex v of T'. Let D be a $\gamma_h(T)$-set. If $v_3 \notin D$, then $S_{v_3} \subset D$. If $S_{v_3} \subset D$, then $D \cup \{v_3\} - S_{v_3}$ is an HDS of T with $|D \cup \{v_3\} - S_{v_3}| < |D|$, which is a contradiction. So, $v_3 \in D$. It is easy to see that $D - \{v_3\}$ is an HDS of T'. So, $\gamma_h(T') \leq \gamma_h(T) - 1$, i.e., $\gamma_h(T') \leq m$. Therefore, by the induction hypothesis $T' \in \mathcal{T}_c$. So, there is a $\gamma_h(T')$-set D' such that D' induces a connected subtree of T'. Because of $diam(T') \geq 5$, $v_5 \in D'$. Clearly, $v_3 \notin D'$ and so $D' \cup \{v_3\}$ is a $\gamma_h(T)$-set, that is, $\gamma_h(T) \leq \gamma_h(T') + 1$. This, together with $\gamma_h(T') \leq \gamma_h(T) - 1$, implies that $\gamma_h(T') = m$. Since $T' \in \mathcal{T}_c$, by Observation 1 there is a $\gamma_{hR}(T')$-function $f' = (V_0^{f'}, V_1^{f'}, V_2^{f'})$ with $V_1^{f'} = \emptyset$ such that $V_2^{f'}$ induces a connected subtree of T'. Since $\gamma_h(T') = m$, we have $\gamma_{hR}(T') = 2m$. If $f'(v_4) = 0$, then the subtree of T' induced by $V_2^{f'}$ is a disconnected tree. So, we have $v_4 \in V_2^{f'}$. Then $f = (V_0^{f'} \cup S_{v_3}, \emptyset, V_2^{f'} \cup \{v_3\})$ is an HRDF on T. So, $\gamma_{hR}(T) \leq w(f) = 2m+2$. On the other hand, let g be a $\gamma_{hR}(T)$-function. Since $|S_{v_3}| \geq 2$ and $|S_{v_5}| \geq 2$, we may assume $g(v_3) = g(v_5) = 2$. Let g' be the restriction of g on T' with $g'(v_3) = 0$. Then g' is an HRDF on T'. So, $\gamma_{hR}(T') \leq \gamma_{hR}(T) - 2$. It follows that $\gamma_{hR}(T) = 2m+2$, and, therefore, T is a hop Roman tree. Since $f = (V_0^f = V_0^{f'} \cup S_{v_3}, V_1^f = \emptyset, V_2^f = V_2^{f'} \cup \{v_3\})$ is a $\gamma_{hR}(T)$-function with $V_1^f = \emptyset$

Algorithm 2.2: Connected-Hop-Roman-Tree (T)

```
Input: A tree T.
Output: Yes: if T ∈ 𝒯_c; otherwise, NO.
1  Let d be the diameter of T. If d ∈ {0,1,2}, then return NO; if d ∈ {3,4}, then return YES.
2  I(T) := Compute-Inner-Vertices (T) ;
3  for each v ∈ I(T)   |S_v| := 0 ;
4  for each x ∈ N(v)   find := true ;
5  for each y ∈ N(x)\{v}     if  y ∈ I(T) then
6  |  find := false ;
7  end
8  if  find then
9  |  |S_v| = |S_v| + deg(x) − 1 ;
10 end
11 if  |S_v| < 2 then
12 |  return NO;
13 end
14 return YES;
```

such that V_2^f induces a connected subtree of T, it follows that $T \in \mathcal{T}_c$. This completes the proof. □

Lemma 3. *Let T be a tree. Algorithm 2.2 decides whether $T \in \mathcal{T}_c$ in $\mathcal{O}(|V(T)|)$ time.*

Proof. If the diameter of T is in $\{0, 1, 2\}$, then $T \notin \mathcal{T}_c$ and if the diameter of T is in $\{3, 4\}$, then $T \in \mathcal{T}_c$. So, if $diam(T) \geq 5$, then Algorithm Connected-Hop-Roman-Tree works correctly. Let $diam(T) \geq 5$. Let v be any inner vertex of T. Recall that $S_v = N_2'(v) - \cup_{u \in I(T)\setminus\{v\}} N_2'(u)$, where $N_2'(v) = \{u \in V(T)\setminus I(T) | d(u, v) = 2\}$. Let $x \in N(v)$ and let $y \in N(x)\setminus\{v\}$. Clearly, $y \in N_2(v)$. Clearly, if $y \in I(T)$, then $w \notin S_v$ for any $w \in N(x)$. If there is $u \in I(T)$ such that $y \in N_2(u)$ and $d(u, v) = 4$, then $y \in I(T)$.So, for computing S_v it suffices to check whether there is an inner vertex of T in $N(x)\setminus\{v\}$. If all vertices in $N(x)\setminus\{v\}$ do not belong to $I(T)$, then $N(x)\setminus\{v\} \subseteq S_v$. If there is an inner vertex of T in $N(x)\setminus\{v\}$, then any vertex in $N(x)\setminus\{v\}$ does not belong to S_v. It is easy to see that Algorithm Connected-Hop-Roman-Tree does this.

Assume that Algorithm Connected-Hop-Roman-Tree considers $v \in I(T)$ and $x \in N(v)$, where $I(T)$ is the set of all inner vertices of T. If all vertices in $N(x)\setminus\{v\}$ do not belong to $I(T)$, then the condition in line #8 of Algorithm Connected-Hop-Roman-Tree does not hold for all vertices in $N(x)\setminus\{v\}$. It follows that the variable $find$ has the value *true*. So, Algorithm Connected-Hop-Roman-Tree executes the instruction " $|S_v| = |S_v| + \deg(x) - 1$ ", i.e., Algorithm Connected-Hop-Roman-Tree add $N(x)\setminus\{v\}$ to S_v. If there is an inner vertex of T in $N(x)\setminus\{v\}$, then the condition in line #8 of Algorithm Connected-Hop-Roman-Tree holds for at least one vertex in $N(x)\setminus\{v\}$. It follows that the variable $find$ has the value $false$. So, Algorithm Connected-Hop-Roman-Tree does not execute the instruction "

$|S_v| = |S_v| + \deg(x) - 1$ ", i.e., Algorithm Connected-Hop-Roman-Tree does not add any vertex of $N(x)\backslash\{v\}$ to S_v.
It remains to compute the time complexity of Algorithm Connected-Hop-Roman-Tree. Let $V(T) = \{v_1, v_2, \ldots, v_n\}$. It is easy to see that the running time of Algorithm Connected-Hop-Roman-Tree is at most $|S_T| = |V(T_{v_1})| + \ldots + |V(T_{v_n})|$. We know by the proof of Lemma 1 that $|S_T| = \mathcal{O}(n)$. This completes the proof. □

By Lemma 3 we have the following result.

Theorem 2. *Given a tree T, there is an optimal algorithm that decides whether $T \in \mathcal{T}_c$.*

We close with the following problem.
Problem 1: Does there exist an algorithm that decides whether $\gamma_{hR}(T) = 2\gamma_h(T)$ for a given tree T?

References

[1] M.P. Álvarez-Ruiz, T. Mediavilla-Gradolph, S.M. Sheikholeslami, J.C. Valenzuela-Tripodoro, and I.G. Yero, *On the strong roman domination number of graphs*, Discrete Appl. Math. **231** (2017), 44–59.

[2] M. Atapour, S.M. Sheikholeslami, and L. Volkmann, *Global Roman domination in trees*, Graphs Combin. **31** (2015), no. 4, 813–825.

[3] S.K. Ayyaswamy, B. Krishnakumari, C. Natarajan, and Y.B. Venkatakrishnan, *Bounds on the hop domination number of a tree*, Proceedings-Mathematical Sciences **125** (2015), no. 4, 449–455.

[4] S.K. Ayyaswamy and C. Natarajan, *Hop domination in graphs*, Manuscript.

[5] E.J. Cockayne, P.A. Jr. Dreyer, S.M. Hedetniemi, and S.T. Hedetniemi, *Roman domination in graphs*, Discrete Math. **278** (2004), no. 1, 11–22.

[6] A. Hansberg and L. Volkmann, *Upper bounds on the k-domination number and the k-Roman domination number*, Discrete Appl. Math. **157** (2009), no. 7, 1634–1639.

[7] T.W. Haynes, S.T. Hedetniemi, and P.J. Slater, *Fundamentals of domination in graphs*, Marcel Dekker, New York, 1998.

[8] M.A. Henning and N. Jafari Rad, *On 2-step and hop dominating sets in graphs*, Graphs Combin. **33** (2017), no. 4, 913–927.

[9] N. Jafari Rada and E. Shabanib, *On the complexity of some hop domination parameters*, Elect. J. Graph Theory Appl. **7** (2019), no. 1, 74–86.

[10] C. Natarajan and S.K. Ayyaswamy, *Hop domination in graphs-ii*, An. Stt. Univ. Ovidius Constanta **23** (2015), no. 2, 187–199.

[11] C.S. ReVelle and K.E. Rosing, *Defendens imperium Romanum: a classical problem in military strategy*, Amer Math. Monthly **107** (2000), 585–594.

[12] E. Shabani, *Hop Roman domination in graphs*, Manuscript (2017).

[13] E Shabani, N Jafari Rad, and A Poureidi, *Graphs with large hop Roman domination number*, Computer Sci. J. Moldova **27** (2019), no. 1, 1–20.

[14] I. Stewart, *Defend the Roman empire!*, Sci. Amer. **281** (1999), no. 6, 136–138.

[15] L. Volkmann, *Signed total Roman domination in digraphs*, Discuss. Math. Graph Theory **37** (2017), no. 1, 261–272.

[16] ______, *The signed total Roman k-domatic number of a graph*, Discuss. Math. Graph Theory **37** (2017), no. 4, 1027–1038.